

Enginyeria del programari I

Benet Campderrich Falgueras
Recerca informàtica, SL

6 crèdits
XP00/05007/00298

Universitat
Oberta
de Catalunya

Informàtica



**Benet Campderrich
Falgueras**

Doctor enginyer industrial.
Especialitzat en enginyeria
del programari i bases de dades.
Professor titular de la
Universitat Rovira i Virgili.

Primera edició:
© Fundació per a la Universitat Oberta de Catalunya
Av. Tibidabo, 39-43, 08035 Barcelona
Disseny: Manel Andreu
Dipòsit legal: B-37.646-00
ISBN: 84-8429-121-9

Cap part d'aquesta publicació, incloent-hi el disseny general i de la coberta, no pot ser copiada, reproduïda, emmagatzemada o transmesa de cap manera ni per cap mitjà, tant si és elèctric, com químic, mecànic, òptic, de gravació, de fotocòpia, o per altres mètodes, sense l'autorització prèvia per escrit dels titulars del copyright.

Introducció

L'enginyeria del programari comprèn els mètodes i les tècniques que es fan servir en el desenvolupament professional de programari; és un camp molt ampli, del qual aquesta assignatura només preveu una part. De les tècniques de programació, se'n tracta en diverses assignatures que els estudiants cursen abans que aquesta. Els temes que no formen part de l'elaboració de programari en sentit estricte, com ara la gestió de projectes i la prova, s'estudien en una altra assignatura, i també el disseny de bases de dades. ⚠

L'enginyeria del programari consta principalment de dues famílies de tècniques:


- Les estructurades, cronològicament les més antigues, que s'estudien en una assignatura posterior.
- Les orientades a objectes (OO), que constitueixen la part principal d'aquesta assignatura, amb les exclusions esmentades.

La resta de l'assignatura és una introducció a l'enginyeria del programari, una introducció necessària, ja que és la primera assignatura que s'estudia en la qual es considera l'enginyeria del programari des d'una perspectiva general.

La tecnologia d'elaboració de programari orientat a objectes ha tingut la major part del seu desenvolupament de 1985 ençà. Com sol passar amb cada nova tecnologia, al principi apareixen moltes tècniques alternatives i el pas del temps comporta que moltes d'aquestes s'abandonin i només en resti una o algunes que esdevenen un estàndard, oficial o de fet. Sembla que la tecnologia orientada a objectes ha arribat fa molt pocs anys a aquesta situació, si més no pel que fa al model bàsic, del qual hi ha l'estàndard anomenat *UML*, ja molt àmpliament acceptat i utilitzat; aquest és el model que fem servir. ⚠

A més del model utilitzat, un altre aspecte fonamental del desenvolupament de programari és el mètode, ja que si ens servim del mateix model ens podem imaginar molts mètodes diferents que empren les notacions del model en un altre ordre o per a altres propòsits diferents. A diferència del cas del model bàsic, és poc probable que s'imposi un mètode com a estàndard; a tot estirar es pot establir com a estàndard legal la presentació d'una determinada documentació elaborada segons un model fixat, però la manera com es treballi per a elaborar-la sempre tindrà un marge de llibertat ampli.

En aquesta assignatura s'ha optat per seguir un mètode inspirat en l'anomenat *Rational Unified Process*, perquè aquest és bastant elaborat i coherent i, a més, ha estat desenvolupat essencialment pel mateix equip que va desenvolupar *UML*, fet que dona una certa garantia de coherència entre el mètode i el model.

Es pot considerar que aquesta assignatura té una part bàsica, un nucli i una cloenda: 

- La part bàsica comprèn uns coneixements fonamentals, que són el mòdul “Introducció a l’enginyeria del programari OO”, dedicat als conceptes generals de l’enginyeria del programari; i el mòdul “UML (I): el model estàtic” i el mòdul “UML (II): el model dinàmic i d’implementació”, dedicats a UML.
- El nucli de l’assignatura el constitueixen els mòduls “Recollida i documentació de requisits”, “Anàlisi orientada a objectes” i “Disseny orientat a objectes”, dedicats a la recollida i documentació de requisits, l’anàlisi i el disseny, respectivament.
- La cloenda és el mòdul “Introducció al programari distribuït”, que dóna una visió general de la tecnologia del programari distribuït orientat a objectes, tema que dóna ocasió d’introduir no solament la problemàtica i les tècniques directament lligades a la distribució del programari, sinó també conceptes que tenen un camp d’aplicació més ampli.

Objectius

L'objectiu general de l'assignatura és que els estudiants aprengueu a desenvolupar programari orientat a objectes, a partir d'uns coneixements previs sòlids sobre la programació orientada a objectes; aquest objectiu es concreta en els següents:

- 1.** Adquirir un coneixement general de l'enginyeria del programari i la seva problemàtica.
- 2.** Assimilar el concepte de *cicle de vida* en general i de *cicle de vida iteratiu i incremental*, amb el del Rational Unified Process com a exemple.
- 3.** Conèixer UML com a model general orientat a objectes.
- 4.** Aprendre a dur a terme la recollida de requisits, l'anàlisi i el disseny de programari en forma orientada a objectes emprant UML.
- 5.** Aprendre a fer l'anàlisi de les tasques dels usuaris i el disseny de la interfície d'usuari.
- 6.** Adquirir una visió de tipus introductori de la tecnologia d'objectes distribuïts.

Continguts

Mòdul didàctic 1

Introducció a l'enginyeria del programari OO

Benet Campderrich Falgueras (0,3 crèdits)

1. Què és l'enginyeria del programari
2. El cicle de vida del programari
3. Desenvolupament estructurat i desenvolupament OO
4. Les eines CASE
5. L'OMG i l'UML

Mòdul didàctic 2

UML (I): el model estàtic

Benet Campderrich Falgueras; Recerca Informàtica, SL (0,8 crèdits)

1. Concepte de model estàtic i de diagrama de classes
2. Classificadors
3. Paquets
4. Classe i conceptes afins
5. Representació dels objectes
6. Relacions entre classes
7. Comentaris i restriccions

Mòdul didàctic 3

UML (II): el model dinàmic i d'implementació

Benet Campderrich Falgueras; Recerca Informàtica, SL (0,8 crèdits)

1. El diagrama d'estats
2. El diagrama de casos d'ús
3. Els diagrames d'interacció
4. El diagrama d'activitats
5. Els diagrames d'implementació

Mòdul didàctic 4

Recollida i documentació de requisits

Benet Campderrich Falgueras (0,6 crèdits)

1. Els requisits
2. Fonts d'informació
3. Passos de la recollida i documentació de requisits
4. La recollida i documentació de requisits de la interfície d'usuari
5. Exemple

Mòdul didàctic 5

Anàlisi orientada a objectes

Benet Campderrich Falgueras; Recerca Informàtica, SL (1,5 crèdits)

1. El paper de l'anàlisi
2. Paquets d'anàlisi i paquets de serveis
3. Revisió dels casos d'ús
4. Especificació de les classes d'anàlisi
5. Especificació formal dels casos d'ús
6. Anàlisi de la interfície d'usuari
7. Exemple

Mòdul didàctic 6

Disseny orientat a objectes

Benet Campderrich Falgueras; Recerca Informàtica, SL (1,5 crèdits)

1. El paper del disseny
2. La reutilització
3. El disseny arquitectònic
4. Disseny dels casos d'ús
5. Revisió del diagrama estàtic de disseny
6. Disseny de la persistència
7. Disseny de la interfície d'usuari
8. Disseny dels sistemes
9. Exemple

Mòdul didàctic 7

Introducció al programari distribuït

Benet Campderrich Falgueras (0,5 crèdits)

1. Entorns distribuïts i entorns oberts
2. Entorns client / servidor clàssics
3. Entorns amb *middleware*: CORBA
4. RMI
5. Documents compostos distribuïts: DCOM
6. Desenvolupament de programari distribuït

Bibliografia

Bibliografia bàsica

Booch, G.; Rumbaugh, J.; Jacobson, I. (1999). *UML. El lenguaje unificado de modelado. Guía del Usuario*. Madrid: Addison-Wesley.

Buschmann, F.; Meunier, R.; Rohnert, H.; Sommerlad, P.; Stal, M. (1996). *A System of Patterns. Pattern-Oriented Software Architecture*. Addison-Wesley.

Gamma, E.; Helm, R.; Johnson, R.; Vlissides, J. (1995). *Design Patterns: Elements of Reusable Object Oriented Software*. Reading: Addison-Wesley.

Jacobson, I.; Booch, G.; Rumbaugh, J. (2000). *El proceso unificado de desarrollo de software*. Addison-Wesley.

Kruchten, P. (2000). *The Rational Unified Process. An Introduction* (2a. ed.). Addison-Wesley.

Larman, C. (1998). *Applying UML and Patterns. An Introduction to Object-Oriented Analysis and Design*. Upper Saddle River: Prentice-Hall.

Orfali, R.; Harkey, D.; Edwards, J. (1999). *Client/Server Survival Guide* (3a. ed.). Nova York: John Wiley & Sons.

Richter, Ch. (1999). *Designing Flexible Object-Oriented Systems with UML*. Indianapolis: MacMillan.

Rosenberg, D.; Scott, K. (1999). *Use Case Driven Object Modeling with UML: A Practical Approach*. Addison-Wesley.

Rumbaugh, J.; Jacobson, I.; Booch, G. (2000). *UML. El lenguaje de modelado unificado. Manual de referencia*. Addison-Wesley.

Slama, D.; Garbis, J.; Russell, P. (1999). *Enterprise CORBA*. Prentice Hall.

Weinschenk, S.; Jamar, P.; Yeo, S.C. (1997). *GUI Design Essentials*. John Wiley & Sons.

Bibliografia complementària

Coplien, J.O.; Schmidt, D.C. (1995). *Pattern Languages of Program Design*. Addison-Wesley.

Coplien, J.O.; Schmidt, D.C.; Vlissides, J.M.; Kerth, N. (1996). *Pattern Languages of Program Design 2*. Addison-Wesley.

Eriksson, H.E.; Penker, M. (1997). *UML Toolkit*. John Wiley & Sons.

Fowler, M.; Scott, K. (1999). *UML gota a gota*. Amsterdam: Prentice-Hall.

Harrison, N.; Foote, B.; Rohnert, H. (1999). *Pattern Languages of Program Design 4*. Addison-Wesley.

Jacobson, I. (1994). *Object-Oriented Software Engineering: A Use Case Driven Approach*. Addison-Wesley.

Martin, R.C.; Riehle, D.; Buschmann, F.; Vlissides, J.M. (1997). *Pattern Languages of Program Design 3 (Software Patterns Series)*. Addison-Wesley.

Meyer, B. (1999). *Construcción de software orientado a objetos* (2a. ed.). Madrid: Prentice-Hall.

Pressman, R.S. (1997). *Ingeniería del software. Un enfoque práctico* (4a. ed.). Madrid: McGraw-Hill.

Rock-Evans, R. (1998). *DCOM Explained*. Digital Press.

Sommerville, I. (1995). *Software Engineering* (5a. ed.). Harlow: Addison-Wesley.

